Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2006 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2006

Conceptual Map of Software Environmental Impact

J. Pinto
Universidad Simón Bolívar

Maeía Pérez
Universidad Simón Bolívar

Luis Mendoza
Universidad Simón Bolívar

A. Grimán
Universidad Simón Bolívar

M. Kräuter
Universidad Simón Bolívar

Conceptual Map of Software Environmental Impact

Pinto, J.
Processes and Systems Department – LISI
Universidad Simón Bolívar
Caracas – Venezuela
joseepinto@cantv.net

Pérez, M.
Processes and Systems Department – LISI
Universidad Simón Bolívar
Caracas – Venezuela
movalles@usb.ve

Mendoza, L.
Processes and Systems Department – LISI
Universidad Simón Bolívar
Caracas – Venezuela
lmendoza@usb.ve

Grimán, A.
Processes and Systems Department – LISI
Universidad Simón Bolívar
Caracas – Venezuela
agriman@usb.ve

Kräuter, M.
Processes and Systems Department
Universidad Simón Bolívar
Caracas – Venezuela
mkrauter@usb.ve

ABSTRACT

Nowadays, there is an increasing necessity for renewing hardware platforms, mainly because of the extended requirements of software products. On the other hand, pollution and health problems are being experienced by software users resulting from the replacement and use of computers. Taking into account the global concerns about damage to the ecosystem, enough arguments are found to develop a software quality model that includes the estimation of environmental impact. However, the fundamental concepts to be considered are not clear yet. This work is part of a research in progress; it proposes a map including concepts and relationships, which intends to establish the scope of a model (based on ISO/IEC 9126) for estimating system quality supported by a software eco-design. This set of concepts and relationships will increase conceptual maturity of the subject and the relevant relationships in it.

Keywords

Environment, Internal Environmental Audit, Quality Model

INTRODUCTION

When people talk about the environment and the damage to which it is continuously subjected, they think that this damage is caused by the production of industrial and domestic waste, the use of fertilizers and aerosols, carbon monoxide emissions, tree cutting and burning, etc. However, one could ask: could the use of software products have some kind of environmental impact?

(Sommer, 2005) states that a considerable amount of electronic components is being received and recycled in Asia, which are in such condition that the health of workers, communities, and the environment results endangered. The same author adds that due to the contact with electronic waste, these workers are exposed to the danger derived from heavy metals, which leave lethal residues in the body, soil and water streams.

The growing need to renew the hardware platform of our computers, which results from the more and more demanding requirements of software products, and the subsequent pollution produced by the replaced parts, together with the health problems suffered by their users and the general concern for the damage to the ecosystem, are only some of the reasons why the environmental impact caused by this kind of products should be studied.

It is important to note that the study of aspects such as hardware flexibility for satisfying software product requirements are not part of the goal of this work which is focused on the software products.

This paper presents the conceptual bases of a research in progress intended to evaluate to which extent a software quality model based on ISO-9126 favors the eco-design of software products and, if necessary, to incorporate improvements considering this issue. In the context of this article, eco-design will be considered like the design of a software which has the objective to obtain the smaller environmental impact when software is in its operation environment. This research is being conducted applying a methodology that is an adaptation from the Systemic Methodological Framework (Pérez, Grimán, Mendoza, and Rojas, 2004) for Information Systems Research, which is based on the Research-Action method (Baskerville and Pries, 1996) and the DESMET methodology (Kitchenham, 1996).

SOFTWARE QUALITY

To find a way to evaluate or understand the environmental impact of a software product, it is necessary to understand some concepts such as software product, software quality, quality models, and environment.

According to (Pressman, 2002), *software quality* is the conformance to explicitly defined functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of professionally developed software

This definition could also be applied to *software product quality*, inasmuch as this is a particular kind of software. It should be considered that, according to (ISO/IEC, 2001a), *software product* is a set of computer programs, procedures, data, plus the related documentation.

(ISO/IEC, 2001a) also states that a *quality model* constitutes a set of characteristics and the relationships among them, which provide the bases to specify quality requirements and assess quality. According to the same authors, software product quality should be evaluated using a defined quality model.

Before explaining what could be the environment of a software product as a whole system, it is convenient to define environment and organization. The COVENIN-ISO 14050:1999 standard by (Fondonorma, 1999) defines *environment* as the setting where an organization operates, including water, air, earth, natural resources, fauna, human beings, and the interrelationships among them. In addition, the same standard defines *organization* as any company, corporation, firm, enterprise, entity or institution, association or any part of them, in the form of a society or not, public or private, with its own functions, administration and management.

A software product is part of an organization and, accordingly, this latter could be the environment where the product operates. However, and due to its general nature, one of the difficulties this approach may face is the definition of the actual boundaries of the system to which the software product belongs. (Johansen, 2000), speaking of systems in general, points out that the environment of a system will be determined by the problem the researcher is studying and, therefore, a way of doing this it is by setting up the actual boundaries of the system according to the concrete problem. Churchman, cited by (Johansen, 2000), states that something will be part of the environment of a system if this latter cannot face that something and if that something is important for its objectives.

ENVIRONMENTAL IMPACT

Considering what has been pointed out before, environment for a software product could be defined as its operational setting and should include all elements that could affect its objectives in one way or another, being the software unable to do anything regarding how it was defined. These elements could include people, energy and hardware.

Due to the complexity that means defining the actual boundaries of the software product environment and considering the work by (Quella and Schmidt, 2003), determining the environmental impact of this kind of products could imply measuring much more than the energy consumed by the hardware which is required to run them. For instance, (Quella and Schmidt, 2003) state that the e-commerce businesses increase the number of transports and packages required to sell a product. This, undoubtedly, should be assessed in the light of the environmental impact involved.

In addition, there are other environmental issues related to the use of software products, such as headaches, fatigue, and the repetitive stress suffered by users according to (ISO/IEC, 2001b). (Lombarda, w/o year) points out that these issues are giving rise to claims filed by unions in different regions of the world, and have come to such an extent that the International

Labor Organization recommended that exposure time before the computer displays should be limited to no more than 4 hours a day. According to the author, the French Workers Organization, in turn, also reported disorders in eyes, mussels and head, hair loss, general fatigue, irritability, and chest pains. Thanks to these reports, French workers succeeded in having their task declared “unhealthy”, and, as a consequence, their work schedule was reduced to four hours and a half a day.

The relationships between the concepts already described are presented in Figure 1, using UML.

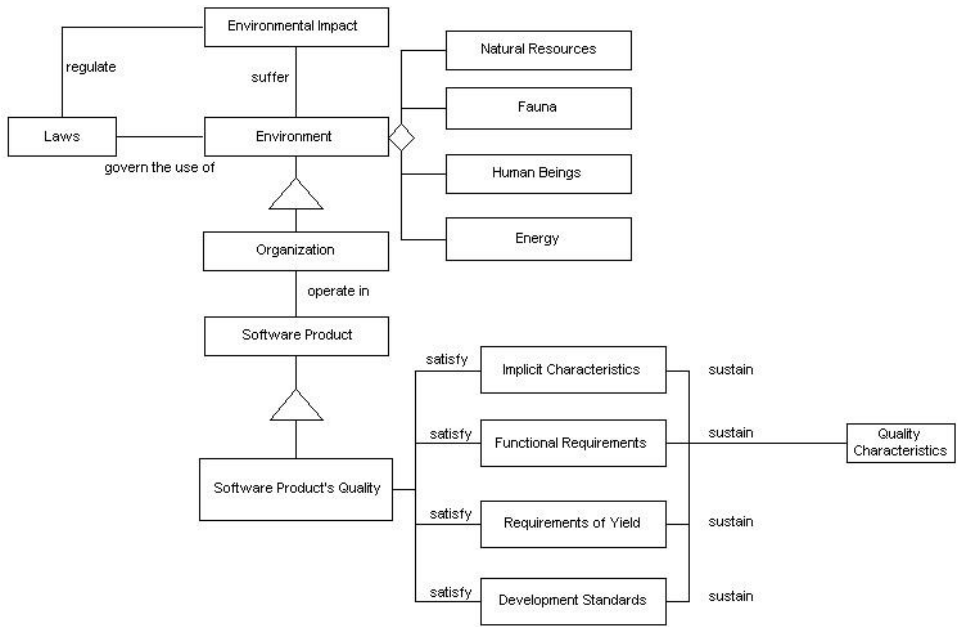

Figure 1. Relationships between environmental impact, environment and software product

SOFTWARE QUALITY AND ENVIRONMENTAL IMPACT

System quality has been represented in multiple models, including FURPS, the models proposed by *Dromey* and *McCall*, and the one developed by *ISO/IEC-9126*. Each one of these standards organize and break down the software's quality attributes. As these attributes do not change over time (Pressman, 2002), it is useful to study them within the framework of the described research currently in progress. It should be remembered that this research aims to study a quality model based on ISO/IEC-9126.

One of the most relevant characteristics of *Dromey’s* model is that it postulates that quality assessment is different for every software product. Therefore, this model proposes a dynamic process and considers the following steps:

- Selection of the high-level attributes to be evaluated.
- Listing of the components or modules of the system.
- Identification of the properties of the components, which affect the attributes selected.
- Determination of the extent to which each property affects quality attributes.
- Assessment of the model and identification of weaknesses.

McCall’s model, according to (Pressman, 2002), states that factors affecting software quality are grouped according to their operational characteristics, their ability to change, and their adaptability to new environments. This author specifically points out that the operation of a product depends on correctness, reliability, usability, integrity, and efficiency; product transition is a function of portability, reusability, and interoperability; and product review is a measure of maintainability, flexibility, and testability.

According to Dromey’s model, each one of the factors already mentioned depends, among others, on the following metrics:

- Reliability: consistency, accuracy, and fault tolerance.

- Efficiency: performance efficiency and storage efficiency

Accordingly, environmental impact could be a function of reliability and efficiency. This could be justified arguing that a system faulty behavior or bad design could lead to an inappropriate use of resources, such as memory, and, subsequently, electric power, not only due to the additional consumption required by storage devices, but also by cooling systems.

According to (Pressman, 2002), *FURPS* is a quality model proposed by Hewlett-Packard, in which the following quality factors are developed: functionality, usability, reliability, performance, and supportability. This model does not consider that environmental impact of a software product is a factor to define its quality. However, and like in Dromey's model, it does consider factors that could be related to this variable, specifically, reliability and performance.

(ISO/IEC, 2001a) states that *ISO/IEC 9126* identifies a set of basic characteristics for every software product. This can be achieved by modeling three different views of the software quality issue: external and internal quality, and use quality. This standard formulates for each one of them a series of quality characteristics. Specifically as to external and internal quality, it proposes six characteristics (functionality, reliability, usability, efficiency, maintainability, and portability), which are subdivided into sub-characteristics.

Use quality considers that quality under this perspective is defined according to the following characteristics: effectiveness, productivity, safety, and satisfaction. Within the framework of this research, it is important to point out that safety definition provided by this standard makes explicit reference to the environment. It specifically states that safety is the ability of the software product to achieve acceptable levels of risks to people, business, software, properties, and even the environment within a specific context.

ISO 14040:1999, *ISO 14041:1998*, *ISO 14042:2000*, and *ISO 14043:2000* aim at identifying and measuring directly or indirectly the environmental impact associated to a product, process or service (International Institute for Sustainable Development, 1996). Due to the fact that, specifically, these standards classify, characterize and evaluate product environmental impact, without excluding the software products, and make reference to aspects such as resource reduction and human health, both characteristics related to the environment, it seems logical to consider aspects related to the environmental variable in all quality software model with the purpose of gaining a software eco-design. Below, a brief description of each one of these standards (ISO, 2002):

- ISO 14040:1999 – Life-Cycle Assessment—Principles and Framework: this standard provides the general principles, working framework, and methodological requirements for the analysis of the life cycle of products and services. It states that a life-cycle analysis comprises the following stages: objective and scope definition, analysis of the inventory of inputs and outputs of the product or service, impact assessment and results interpretation.
- ISO 14041:1998 **-** Life Cycle Assessment—Inventory Analysis: this standard provides a guide to determine the objective and scope of a life cycle and to conduct an inventory of the life cycle.
- ISO 14042:2000 **-** Life Cycle Assessment—Impact Assessment: this standard provides a guide for the assessment phase of the life-cycle impact.
- ISO 14043:2000 **-** Life Cycle Assessment—Interpretation: this standard provides a guide to interpret the results of a study of the assessment of a product or service life cycle.

The model in Figure 2 is obtained when the aspects mentioned in this section are incorporated into the conceptual model already presented. For simplicity purposes, only the quality characteristics considered by ISO/IEC 9126 are included, and only a few standards included in the ISO 14040 series are mentioned.

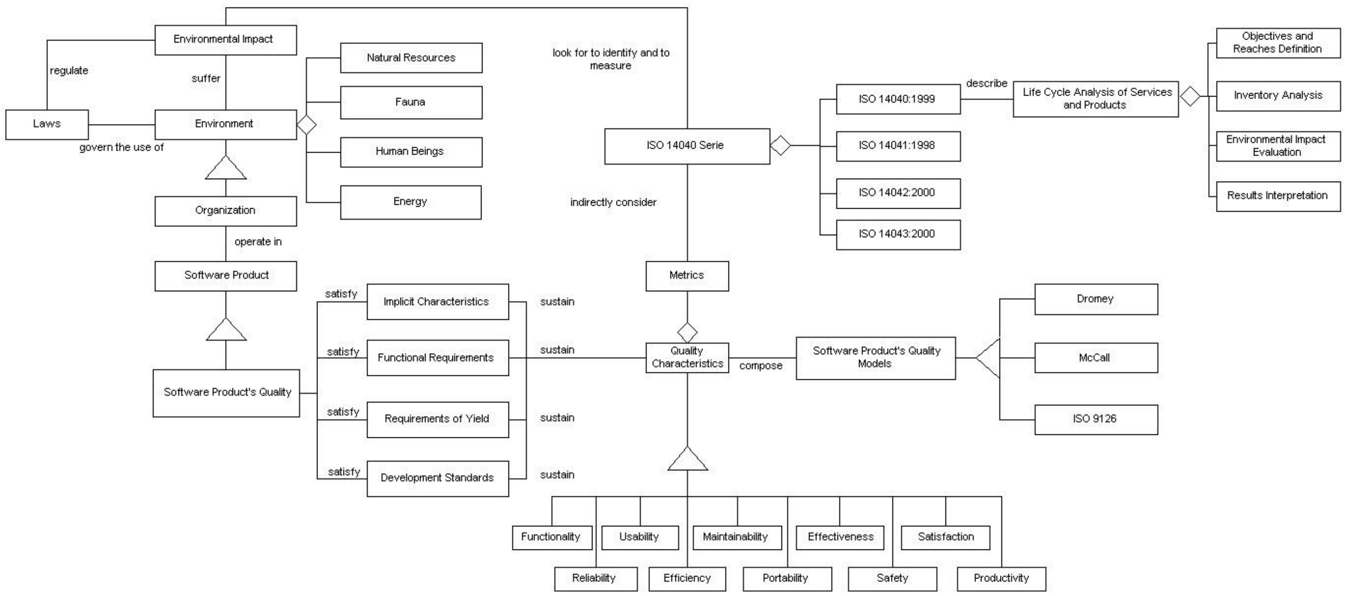

Figure 2. Conceptual Model

CONCLUSIONS AND FUTURE WORKS

Based on the ideas presented before, it could be concluded that a software quality model would provide an eco-design as long as measurable aspects included in the model consider, directly or indirectly, the impact on the environment where the product operates.

Legal regulations provide a reference framework to determine the actual system boundaries and, therefore, the aspects that, in the first instance, should be evaluated by the quality model to develop an eco-design.

In the light of the concepts previously presented and their relationships a software quality model will be submitted to an internal environmental audit to verify if it fosters environmental quality.

REFERENCES

1. Baskerville, R. and Pries, J (1996) Grounded Action Research: A Method for Understanding IT in Practice, *Accounting, Management and Information Technologies*, 9, 1, 1-23.
2. Fondonorma (1999) COVENIN-ISO 14050:1999, Fondonorma, Caracas
3. International Institute for Sustainable Development (1996) Global Green Standards: ISO 14000 and Sustainable Development. Available On-Line: http://www.iisd.org
4. ISO (2002) Application of the ISO 14000 family. Available On-Line: http://www.iso.org
5. ISO/IEC (2001a) ISO/IEC 9126-1,ISO/IEC, Switzerland.
6. ISO/IEC (2001b) ISO/IEC 9126-4.ISO/IEC, Switzerland.
7. Johansen, O. (2000) *Introducción a la Teoría General de Sistemas*. Editorial Limusa.
8. Kitchenham, B. (1996) DESMET: A method for Evaluating Software Engineering methods and tools, *Technical report TR 96-09,* Department of Computer Science, University of Keele, England
9. Lombarda, A. (s/a) Sintomatología y factores del impacto sobre la salud visual. Available On-Line: http://www.ecofield.com.ar/opinion/opi46.htm
10. Mendoza, L., Pérez, M. and Rojas, T. (2005) Prototipo de Modelo Sistémico de Calidad (MOSCA) del Software, *Computación y Sistemas*, 8, 3, 196-217.

11. Pérez, M., Grimán, A., Mendoza, L. & Rojas, T (2004) A Systemic Methodological Framework for IS Research, *Americas Conference on Information Systems AMCIS 2004*, New York, USA. Available On-Line: http://www.lisi.usb.ve/publicaciones
12. Pressman, R., (2002) Ingeniería del Software. Un enfoque práctico. (5ª ed.), McGraw-Hill/Interamericana de España, S. A. U.
13. Quella, F. & Schmidt, W. (2003) Integrating Environmental Aspects into Product Design and Development, The new ISO TR 14062 – Part 2: Contents and Practical Solutions. Available On-Line: http://dx.doi.org/10.1065/ehs2003.02.005
14. Sommer, M. (2005) El lado oscuro de la chatarra electrónica. Available On-Line: http://www.tierramerica.net/2005/0521/grandesplumas.shtml